OPERATIONS ENGINEERING:
APPLYING HANDS-ON EXPERIENCE TO THE DEVELOPMENT PROCESS

Gary Alcott* & Wende Peters**

** Space Data Systems Office, Goddard Space Flight Center,*
Code 510.2, Greenbelt, MD 20771 USA
Fax: (301) 286-9241. E-mail: gary.alcott@gsfc.nasa.gov.usa
***TRW, Global Change Initiatives Division*
7474 Greenway Center Drive, Suite 400
Greenbelt, MD 20770 USA
Fax: (301) 507-5863 E-mail: wende.peters@trw.com

ABSTRACT. This paper presents the concepts behind and benefits of operations engineering as applied to the requirements, design, development, and testing of data processing systems. The goal of operations engineering is to reduce overall life-cycle costs by integrating operations experience with the development process. To achieve this goal, operations engineering seeks to: reduce development costs by assuring operational requirements are incorporated into the design and development process as early as possible and reduce operational costs by decreasing operations staffing requirements and other related costs through improved system capabilities. The areas for improved system capabilities cover: system recovery, data recovery, fault isolation, system operability, system flexibility, system automation, and system reporting. This paper describes how operations engineering is integrated with the development process and discusses difficulties and misconceptions experienced in using operations engineering.

1. OPERATIONS ENGINEERING

Operations engineering (OE) is a segment of systems engineering focused upon incorporating operational experience, knowledge, and insight into the development process to engineer life-cycle costs (LCC). The goals of operations engineering are to reduce LCC throughout development, testing, and operations of a system and to improve system performance and system operability. OE provides a systems level view throughout the entire system life cycle. During each phase of the development cycle, the operations engineer participates in the activities particular to that phase, carrying through the corporate knowledge of each previous phase and maintaining a focus on operability and overall costs.

One of the primary benefits of OE is that it offers a different and somewhat diametrically opposed perspective from that of traditional system development. Developers are necessarily focused on the functional requirements for the system and define their design based on an assumption of nominal operating conditions. As a result, error and boundary conditions are addressed as an addition to the system design. In other words, the developers assume the system will work as designed. Operations engineers must operate on the assumption that the system will not work as anticipated, either as a result of a design flaw or an anomaly within the interfaces driving the system and seek to minimize the operational impact when anomalies occur. This is necessary because most systems optimize their operational staffing profile based on nominal conditions - the operations engineer must tailor the design and implementation of the system to work within that profile while maintaining the capability to recover from internal and external errors. Even when the system performs as specified, circumstances regarding mission support and spacecraft characteristics may change limiting the usefulness of the system. The operations engineer works to include sufficient flexibility into the system design to adapt the operations concept to meet the new circumstances. This position does not imply an assumption that the system design is otherwise incomplete or inadequate. Rather, given highly complex systems with often unstable requirements, it is impossible to account for every operational factor. Assuring that the functional requirements are met, although not always easy, is relatively straight forward; ensuring all the infrastructure needed to support those requirements are met is not always apparent.

During development, operations engineering seeks to guide the development process by providing a goal oriented analysis to reduce labor intensive activity. By involving operations in the development process potential problems can be caught early and therefore reduce or prevent rework later on in the life cycle. Additionally, opportunities for incorporating system enhancements at minimal cost can be taken advantage of through continuous feedback from the operations personnel. During testing, existing operational resources can be used to support the testing effort thereby reducing the required testing staff. Test personnel can concentrate on the aspects specific to testing and leave the operational aspects of testing to the operations personnel. During operations, staffing is reduced as a result of changes made during development, as the operational staff continues to refine the system through sustaining engineering to further reduce operational costs. Essentially, the merits of an operations engineering team is to provide a mechanism to affect the design and development of the system as early as possible with real world operational experience and concerns.

2. OPERATIONAL FACTORS IN SYSTEM DESIGN

With the operations perspective in mind, a large portion of OE is devoted to designing and building an infrastructure within the system that allows the system to recover from failures and anomalies and to prevent them from propagating to other system processes. The primary operational factors to consider are: system recovery, data recovery, fault isolation and analysis, system flexibility, operability, system automation, and system reporting.

System recovery involves recovering from an actual failure of the system itself. This is usually associated with a hardware failure, but could also be a software failure. There are two aspects to system recovery: short term function recovery and system repair. Short term function recovery, which is measured by the system's mean time to restore (MTTRes), represents restoring the lost functionality. Generally this involves failing over to a backup capability and allowing the system to continue to function until repairs can take place. System repair, measured by mean time to repair (MTTR), is the actual repairing or replacement of the failed part, thereby restoring the system to full capability.

Data recovery involves restoring the data on the system to the state it was in prior to failure. To ensure the capability for data restoration, logging functions for input and output data, mission and metadata, must be provided.

Fault isolation and analysis is the ability to identify the cause of a system or data problem based on the observed behavior of the system. To provide for sufficient fault isolation capabilities, the system must provide a mechanism so that each interface and processing point can be “tapped” by an operator. These monitor points provide the capability to extract data for dumping and analysis, as well as a point at which test data can be injected into the system. Tools developed in this area are equally useful to developers, testers, and operators. In addition, the system must maintain an operations log to record the sequence of events that led to a failure. The access to actual data processed as well as the events which occurred concurrently with this processing are the key elements in providing fault isolation capabilities.

System flexibility involves being able to reconfigure the system to support unplanned or extemporaneous activity. Almost all missions require, at various times, support that was unforeseen when the requirements were written and when the system was designed. Many of these instances take the form of special requests from instrument users and other ground system entities. In addition, during pre-launch testing, systems are routinely requested to support test scenarios that exceed operational scenarios defined for the system. How well a system meets these “non-requirements” is usually determined by how much flexibility was designed and built into the system. Many of a system's derived requirements may result from a goal of increased system flexibility. Another aspect of system flexibility is a manual override capability. Since not all contingencies can be planned for in advance, it is necessary to allow the system operator to override any automated system function to account for the unexpected. In addition to being able to override the system, the operator needs to be able to determine the state of the system to properly act on a problem.

System operability is a measure of how easy the system is for an operator to use, directly affecting the level of operations staffing. Function accessibility and screen layouts for manual intervention are important factors in determining how difficult the system is to run. Operational experience is especially important in determining what activities will be manually intensive and which are suitable for automation, as well as, what priority each task should have.

System automation is used to reduce staffing needs and increase the efficiency of the existing personnel. Automation should be applied to all routine functions. If an operator does the same function every time based on information supplied by the system, then the system should be performing the function without operator input. Prime candidates for automation include: system control and monitoring, system failure recovery and reporting, data processing control and monitoring, data evaluation and accounting, data recovery and reporting, and trend analysis. Automated activities should be parameter driven so that they can be adjusted to meet changing operational conditions. A frequent mechanism proposed for introducing automation into a system is the use of expert systems. OE should serve to evaluate the necessity for expert systems as well as their appropriate areas of application. Often, OE will identify conventional software in the form of computer aided analysis tools that are as effective in increasing productivity but better suited to the system needs than more expensive expert systems applications.

System reporting refers to the degree of information feedback and exchange that is provided to the operator via the system. The primary mechanism for this reporting is the system Computer Human Interface (CHI). To support optimized operations, a CHI should be graphics based and provide equal access to system functions from any operator position. Following standard human factors practices, the CHI should be hierarchical, providing increasing levels of detail only upon operator request. Design of the reporting capabilities and the CHI should focus on providing access to all system statistics while responsibly limiting access to system control as appropriate. The goal is to provide an operator with all the facts necessary to intervene during system recovery or fault isolating, without inundating operations personnel during nominal conditions. The flexibility of reporting extends to formatting and presentation method. While some standard reports should be built into the system, it is more important to provide the flexibility to format reports according to the nature of the situation and to permit electronic, printed, or visual displays of the reported information. This permits future operations to respond to special management and trending report requests as well as standard report formats identified early in the design phases. In addition, the operations engineers will identify thresholds for different system reporting parameters, to prevent overload in status messages and to highlight critical levels that must be addressed by operations personnel.

3. REQUIREMENTS FOR OPERATIONS ENGINEERS

To implement OE as a part of the systems development process, operations engineers with the appropriate skill base are needed. Operations engineers provide a unique combination of an operations perspective within engineering disciplines. Thus, an operations engineer must have experience with or exposure to system operations that are representative of the functionality of the system being developed, as well as sufficient knowledge about each stage of the life cycle to recognize appropriate cost drivers and areas to impact. The qualifications for the operations engineer are as follows:

- Experience as an operator, sustaining engineer, or test engineer for an operational system which provides similar capabilities as the proposed system,
- Systems engineering training in the methodology proposed for the development of the system. As this training is routinely provided to the system engineering and development teams, it is vital to plan for and include operations engineering participation in the training budget.

In addition to this base of training and experience, the OE team must be recognized as a matrixed member to each segment of the system organization. As the system progresses through its life cycle, the operations engineer provides technical support to different managers and technical leads. Whenever these transitions occur, it is vital that: (1) the project management define to the project organization the importance of the OE role in that system phase and (2) the phase manager define clearly to the operations engineer what specific goals are to be met through the application of OE to that system phase.

4. OPERATIONS ENGINEERING IN SYSTEM DEVELOPMENT LIFE CYCLE

A large part of operations engineering is to provide the developers with a new perspective, giving them a better understanding of what operations does and why. Developers should view the operations engineer as a developmental resource, to help to aid and provide direction in the development process. Operations engineers also serve as a store of “corporate knowledge” or act as living “lessons learned documents” and can provide insight as to what has and has not worked in the past and what potential problems might arise. This “corporate knowledge” should also be viewed as a source for risk identification, analysis, and mitigation.

OE should begin as early as possible in the system life cycle and should continue throughout the development, testing, operational deployment, and sustaining engineering processes. The earlier OE begins, the earlier problems with the developing system can be identified and fixed before corrections become too costly or cannot be accomplished in the remaining schedule.

Requirements Phase: OE involvement in requirements definition and analysis is primarily to ensure that the general areas of concern mentioned in the previous section are addressed in the requirements and accounted for in development planning. Operations engineers should be responsible for developing the operations concept, projected staffing profiles, and operational scenarios and for deriving operational requirements as driven by the operations concept.

Design Phase: Operations engineers should participate as full members of the systems engineering team to define the high level system design. The OE emphasis during this phase will be to ensure consideration of the identified operational requirements and to guide design trades towards the more operationally appropriate and LCC conscious options. During this phase, the operations engineers also begin production of preliminary end user products such as user's guides, monitor and control screens, and management reports. These products serve as a bottom up approach to engineering, bringing to light any additional derived operational requirements that might not have already been considered. In addition, the operational scenarios are refined to reflect the system design and should not only show the data moving throughout the system from ingest to output, but also include all byproducts and the manual infrastructure needed to support them. Also during this phase, the OE personnel perform "operator engineering," treating operators as a subsystem by allocating requirements to them to clearly delineate the manual activities from the system supported ones.

Development/Coding Phase: One frequent misconception is that OE concludes with the end of the design phase. By their nature most operations requirements are subjective and therefore are more greatly affected by the interpretations of the developers than are the functional or performance requirements. Operations requirements are subjective due to the high-level context in which the requirements are written and the descriptive text that is used to state them. The high level requirements result from the operations functions being dependent on the design of the system. Therefore, the operational requirements are kept intentionally vague so as not to drive a specific implementation. The descriptive text is a result of the general goals associated with the operation functions such as "ease of use", "where practical", and operability. Because operations requirements cannot be explicitly defined for a system that has not yet been implemented, it is essential that they evolve with the system throughout its development. The operations engineer thus assumes the role of assisting the developers in interpretation of operational requirements and further derivation of these requirements.

In addition, during the development phase the operations engineer is utilized to provide input to implementation trades. Frequently detailed implementation choices must be made which result in similar technical outcomes; however, one may represent a much more favorable operations impact than the other. Trade analyses should always take into consideration the existing operational scenarios and should include an operations engineer's evaluation. Specific implementation decisions which necessitate OE input are any regarding the degree and scope of system automation. As the level of automation for a system increases, the need for operational input into the design also increases. The operations engineer can help select which areas are suitable for automation and at what level of automation, as well as identify which forms of automation actually represent greater risk and/or expense than they return in saved operations costs. Furthermore, when project scope and budget are impacted, as often occurs in the midst of development, the operations engineer serves to select the areas most appropriate for scaling back, with minimal impact to the eventual operability of the system.

The operations engineer also supports the detailed development of the CHI during this phase. Ideally, the CHI is designed by the end user, with implementation facilitated by a core OE and software team. Where this is not feasible, the operations engineer represents the end user in guiding the CHI layout.

Integration and Test Phase: During testing efforts, operations personnel should be used to perform all functions that would be normally performed by operations, such as running the system and performing post-test data analysis. This serves two purposes: it allows for a smaller testing staff and provides hands-on training for the operators. The testing staff is reduced by only requiring testers for those aspects unique to the testing effort, such as, writing test plans and procedures, administrating the test, and evaluating the test results. By the very nature of most testing activities, the operators are forced to try to run the system under the worst possible failure conditions. Such training will help them prepare for failures that may occur during normal operations. This process is facilitated by the OE team, who represent the transition of the system from development to operations and are best suited to explain the system characteristics to operations personnel and to assist in the definition of operational procedures. OE activities during this phase include: assisting in the development of test cases to more accurately reflect the operational usage of the system; participating in the integration and test efforts; observing system operations; guiding operational agreements with external interfaces; translating the operational scenarios into users guides, training plans, and operational procedures; and developing operational workarounds to be implemented in response to problems identified during testing.

Operational Deployment and Sustaining Engineering Phase: As the system enters operations, OE works to: assist the operations personnel in using the system including developing any workarounds, identifying unanticipated mission characteristics and system requirements to feed into system upgrades and enhancements, and generating a lessons learned documentation from the entire systems process. Ultimately, OE remains as part of the project through the first year of operations from the last system delivery. In support of sustaining engineering, the OE team assists in the identification of areas representing the best potential for new technology insertion.

5. LESSONS LEARNED IN THE IMPLEMENTATION OF OPERATIONS ENGINEERING

The concept and program presented here for the implementation of OE into the system process arose from lessons learned in evolving this approach as part of the development of NASA GSFC systems, including the Hubble Space Telescope Data Capture Facility (HST DCF) , the Packet Processor Data Capture Facilities (Pacor I and Pacor II), and the Earth Observing System (EOS) Data and Operations System (EDOS). The lessons learned cover two aspects: lessons regarding the method by which OE should be included in the system development process and lessons from the results of OE efforts in the development of NASA systems.

IMPLEMENTING OPERATIONS ENGINEERING

In the early efforts to include operations expertise in the development process, no established methodology for this was in place. Initially, operations personnel from existing systems were invited to sit in on design meetings, in an attempt to educate the operations personnel on the upcoming systems. What evolved was a paradigm where the operations personnel educated the developers on how their systems were actually being used and where automation could really have had an impact on operations costs. As this level of interaction increased and expanded in scope, the following lessons were documented:

<u>It is easier to provide the development perspective to an operations representative than to provide the operations perspective to a developer.</u> Many early efforts were made to have development personnel sit in on operations shifts to observe the system in use and to acquire an understanding of operational issues. However, the time necessary to understand the nuances of operations issues and to appreciate the trades between automation and flexibility was very large. Instead, a single representative or a small team of operations personnel can more effectively review design materials and participate in development efforts,

requiring only minimal training in design methodology, which is likely to be provided to the development team.

The evolution of OE throughout the system life cycle must be mapped from the beginning and clearly established as part of the project planning. During the requirements and design phases, OE is a central portion of the day-to-day project activities. However, in the transition to development/coding, OE shifts to a more consulting and specialized role. It is very easy for the OE team to be left out of development decisions. These instances, where the operations expertise was not considered when initial implementation trades were made, represent the costliest system enhancements and/or operational workarounds encountered.

There will be resistance from traditional designers and developers to the OE inputs. OE represents an assumption that at times systems will fail and that operations personnel will need to step in to troubleshoot these problems and use the system to rectify the resultant problems. This necessarily conflicts with the goals of developers to build “perfect” systems that will not fail. To overcome this resistance, management must acknowledge the changeable nature of mission operations and that there is no way to anticipate all mission needs during the design phase. If OE is presented as a resource of operational knowledge, it will be more easily accepted as an equal discipline in the design and development effort. In addition, the process of including yet another representative in the review processes is frequently seen as an additional schedule impact. While the value-added of including an OE perspective can be documented in retrospect, it is important that project management emphasize the importance of the OE input and to facilitate it to have minimal impact to other project teams. This can be accomplished by: (1) including operations engineers as matrixed team members to the design, development, and test organizations, thus providing those organizations with additional resources, and (2) providing sufficient support and training to the OE team members so that their turnaround on issues can be rapid and effective.

Operational requirements and OE recommendations must be officially documented under the same degree of control as other design factors. Frequently operational issues identified during early phases of the system life cycle are dismissed as “implementation issues” and then later forgotten when implementation trades are performed. A project implementing OE must also implement a mechanism which documents operational issues and carries these through the design and development phase, to make sure they are addressed. While the OE team can be made responsible for this tracking, management must establish the authority of these documented issues as equal to other requirement and discrepancy items which are documented.

OPERATIONAL FACTORS IDENTIFIED THROUGH OPERATIONS ENGINEERING

Separate the development, test, and operational environments wherever possible. Contention for computing resources and test and analysis tools represent one of the largest drain on operations staffing and resources, no matter what the degree of automation provided in the system. Earlier systems provided small pools of spare/backup equipment which were jointly shared between operations personnel troubleshooting anomalies, testing personnel performing verification and acceptance, and development personnel performing unit testing. These resources eventually were in such high demand that a separate management cycle had to be established for their scheduling and each group experienced adverse effects to their schedules and deliveries.

Decouple "on-line" and "off-line" functions in the system design. Operationally, the greatest contention for system resources occurs when troubleshooting is being performed on one set of data while mission processing on other data is ongoing. It is vital that any processing not in the end-to-end processing flow for mission data ("on-line" functions), be separated from other processing ("off-line functions") such as report generation, data analysis, or simulation. OE, in the development of operational scenarios, can provide the delineation between these functional categories, to support the separation of them in the design.

Remove the development of the CHI and the analysis/test tools from the same development schedule as the system application software. User screens, report formats, and analysis tools are intrinsically tied to the detailed design of the system as well as the mission characteristics of the spacecraft being supported. The high level design and requirements for the system functionality are due and frozen long before informed, appropriate decisions about screen layout and mission usage can be made. If these development items are removed from the application critical path and instead handled by an implementation team that includes the operations engineer and operations personnel as well as software experts, the ultimate schedule can still be met with fewer reworks and less resultant requests for corrections and enhancements.

Handle legacy mission requirements separately from new functionality wherever possible. Frequently new systems are envisioned to also assume operational processing for existing missions, to further reduce costs while transitioning to new technology. Even with mission standards, the economies of this type of consolidation come from the centralization of operations staff and statistics collection, not from combining the application processes of old and new missions. Detailed requirements for legacy missions, and thus their external interface formats, were engineered to optimized the original system. Attempts to replace this processing across-the-board with new processing result in: breakage between ground system entities, potential de-optimization of processing for the newer missions, and loss of operational expertise in the handling of the legacy missions. It is more effective to restrict the consolidation of mission support to the infrastructure of a system, with no impact to actual data processing. This preserves the existing interfaces while making it simpler to consolidate operations personnel.

Better engineering of LCC and operational requirements does not imply a less skilled operations team. The application of OE to address operational issues and optimize the implementation of automation will "engineer away" the more mundane and routine tasks in mission handling. The remaining tasks, such as handling error conditions and performing fault isolation, require the most skill. The operations savings realized will be in numbers of people required to operate the system and in a reduction of workarounds and sustaining engineering costs, NOT in the skill class of personnel require to operate the system or the level of training these people will require.

6. SUMMARY

In our experience with applying OE to three generations of systems, we have seen significant improvement in each new system with regards to LCC reduction and operational efficiency and see further potential for improvement in future systems. Staffing profiles for these systems have been reduced significantly from one generation to the next while the number of missions supported increases. These systems have also demonstrated the capability to respond to requirements changes with little or no additional development or impact to operations. Operations personnel are no longer being inundated with mundane scheduling and accounting responsibilities, but rather are actively involved in supporting new mission functionality and improving the system capabilities of both the current and future systems. Still, applying OE has not been an easy task. Operations engineers have had to overcome a significant amount of resistance from traditional developers. This requires an aggressive stance on the part of operations engineers to make themselves heard and a stalwart persistence from management to carry it through the development cycle. However, the developers do eventually recognize the benefits that OE provides and accept and even seek out OE input.